

ZX81 - CHR\$X16 – 05/2005 - wilfrigter

ENHANCED LOW and PSEUDO HIRES VIDEO

This is a new hardware project that provides several enhancements to the operation of a real ZX81.

It is based on an original project of mine circa 1985 in which I build a user defined video character pattern generator that plugged directly into the ZX81 edge connector. That project required 6 chips including a 2K RAM chip. The whole system was completely transparent and compatible with the ZX81 video circuit. In addition to the normal ZX81 non-inverted video pattern for each character, the circuit generated additional tables of video patterns for the inverted characters (CHR\$+128). These were stored in 2K non-volatile RAM which the user had to load from tape. A pushbutton was used to select one of 4 additional tables of 64 lowercase characters, combined with special symbols or game characters that could be shown simultaneously with the original ZX characters on the screen. The additional video pattern tables were mapped over the original 1E00 to 1FFF memory space and no additional software was required, as the fonts were manually selected. The system was totally compatible with any other ZX81 software and hardware. The default font on power up was lower case, which used for program listings, looked kind of odd but blew everyone away.

Two decades later, CHR\$16 was designed with just 3 chips by using an 8K EPROM to store the video patterns and by making the fonts selectable in software. Just a few pokes are needed to select one of 16 complete sets of 64 characters including the original ZX81 set of 64 normal and inverted characters. This circuit adds 14 new sets of pattern tables of your choice organized as 7 sets of 128 unique characters patterns. Always so much simpler in hindsight.

to simulate a NOP which the Z80 loads and executes. During the refresh time of each NOP, the Z80 I register generates the address pointer to the pattern table which is combined with the 6 bit character code and a 3 bit line counter in the ULA to select one byte from the ROM to be loaded into the video shift register. The D7 bit is also loaded into the ULA invert register and inverts the signal shifted from the output of the video shift register at 6.5M bits per second.

The CHR\$X16 circuit works as follows:

As the ZX81 generates a live video screen, the Display File at (400C) is scanned by sequentially loading each valid character code in the 8 bit 74HC374 register of the new circuit on the rising edge of each RD. This is followed by the ULA pulling the data bus low to simulate a NOP which the Z80 loads and executes. During the refresh time of each NOP, the Z80 I register generates the address pointer to the pattern table which is combined with the 7 bit character code and the 3 bit line counter of the circuit (74HC192) to select one byte from the new EPROM patterns to be loaded into the video shift register. The D7 bit is also loaded into the ULA invert register and inverts the signal shifted from the output of the video shift register at 6.5M bits per second.

THE EPROM

The 27C64 EPROM is mapped over the entire 8K ROM area. This is ok since the EPROM only responds to /RFSH low, while the internal ROM only responds an active low /RD line.

In order, to select a given pattern table the Z80 Interrupt register must be loaded with the upper 8 bits of the starting address of one of eight pattern tables. This requires setting up a 1 REM line with 5 spaces and few pokes to load the 5 byte ML program.

```
LD A , n
```

```
LD I , A
```

```
RET
```

which is then executed with RAND USR 16514 to select the table. Afterwards a single poke of 16515, $n \times 4 + 224$, (where table $n=0$ to 7) and the RAND USER 16514 is all that is needed to change tables on the fly.

Although each table has 128 characters and not the ideal 256, a special table for pseudo hires programs can be added with significant improvement of “resolution” and uniform code to pattern conversions.

Each table is 1024 bytes with the second 512 block selected by D7 of the CHR\$ code. This requires that the second half of each table uses complemented data as this will in turn be complemented by the ULA at the video shift register output.

The ZX81 default table will start at address 1C00 and end at 1FFFF. The ZX81 printer will continue to use the normal ROM patterns which is no compromise as other patterns, fonts and hires graphics have always been available for the ZX printer.

THE CHR\$ REGISTER

The 74HC374 octal clocked register is used to capture the CHR\$ codes from DFILE to be used as part of the pattern lookup address. Unlike the ZX81, this register also captures D7 and as part of the address which doubles the size of each pattern table. The register is loaded on the rising edge of RD and the tristate output is permanently enabled.

THE LINE COUNTER

The 74HC192 is a pre-settable 4 bit counter of which 3 bits are used to select which 1 of 8 lines of patterns is accessed. Each of the 24 rows of characters is made up of 8 lines of 24 pattern bytes. The counter is pre-set to all ones with the NMI line on the PE pin. The NMI generator is turned off at the start of the screen display. Since the first line of the screen display consists of just a N/L (HALT opcode), this causes the counter to increment to zero which is the starting line for the first character row. The subsequent N/L characters at the end of each line, each cause a HALT, each of which clocks the counter to generate the 3 low order pattern table address bits.

THE ADDRESS DECODER

The high order address decoder uses 3 diodes and a resistor to generate a low chip enable for the EPROM CE pin when A13-15 are low. A low level on the RFSH line enables the EPROM data outputs to be loaded into the video shift register.

THE ROMCS DECODER

The ROM must be disabled when the EPROM is active so an emitter follower is used to force the ROMCS line high when the RD line is high, which occurs when the RFSH line is low.

Any general purpose NPN transistor can be used for this application which minimizes the loading of the RD line.

ASSEMBLY

The circuit is assembled on a perforated board with a feed through female and male connector and is attached inline with the ZX81 PCB edge connector.

FIRMWARE

The ROM patterns are readily available from HIRES programs and from font tables used for the ZX printer. I will combine the best into one EPROM image and the file will be shortly added to this article. Pre-programmed EPROMs will be available to anyone who would like to build this for the price of posting it.

Enjoy

wilf